



Strathprints Institutional Repository

Bezhanishvili, N. and Kupke, C. and Panangaden, P. (2012) *Minimization via duality*. In: Logic, Language, Information and Computation. Lecture notes in computer science / Theoretical Computer Science and General Issues, 7456 LNCS . Springer, pp. 191-205. ISBN 9783642326202

Strathprints is designed to allow users to access the research output of the University of Strathclyde. Copyright © and Moral Rights for the papers on this site are retained by the individual authors and/or other copyright owners. You may not engage in further distribution of the material for any profitmaking activities or any commercial gain. You may freely distribute both the url (<http://strathprints.strath.ac.uk/>) and the content of this paper for research or study, educational, or not-for-profit purposes without prior permission or charge.

Any correspondence concerning this service should be sent to Strathprints administrator: <mailto:strathprints@strath.ac.uk>

Minimization via Duality

Nick Bezhanishvili¹, Clemens Kupke², and Prakash Panangaden³

¹ Department of Computing, Imperial College London

² Department of Computer Science, University of Oxford

³ School of Computer Science, McGill University

Abstract. We show how to use duality theory to construct minimized versions of a wide class of automata. We work out three cases in detail: (a variant of) ordinary automata, weighted automata and probabilistic automata. The basic idea is that instead of constructing a maximal quotient we go to the dual and look for a minimal subalgebra and then return to the original category. Duality ensures that the minimal subobject becomes the maximally quotiented object.

1 Introduction

The main goal of this paper is to exploit a simple observation of category theory that yields some striking results when applied to particular instances. The simple observation is this: a quotient construction in a category is the same as a subobject construction in the opposite category. How can such a simple observation be useful or interesting? First, there is the mathematically nontrivial task of giving a useful alternative description of the opposite category. Second it is useful because finding the relevant subobjects may be easier than an explicit quotient construction in the original category of interest. Furthermore one gets completely new algorithms for the minimization problem. Third, it is interesting because one ties together seemingly unrelated ideas into a coherent whole. These ideas have appeared not only in minimization but in questions related to machine learning, specifically the problem of modelling systems with hidden state [9]

We are going to exploit duality or, more precisely, a dual equivalence for the minimisation of automata, ie. we use the fact that a given category of transition systems, say \mathcal{C} , is *equivalent* to the opposite of another category \mathcal{D}^{op} . Our basic strategy for minimization is as follows. We start with a transition system S of a certain type. We go to the dual category and look for a special subobject: a so-called *zero-generated subobject* which has the property that it *must embed* into any other subobject of the dual object. Then we come back to the original category and find that the zero-generated subobject is the *minimal realization* of the original object. The general abstract statement is of course very easy, practically an observation. Examples of this phenomenon are, however, very interesting.

The first example that we treat in this way is based on reasoning about systems with hidden state [9]. It is related in spirit, but not in detail, to an old algorithm due to Brzozowski [7] which is a minimization algorithm that seems to work by black magic. A precise categorical treatment of Brzozowski's algorithm has recently been given by Bonchi et al. [6]. Our second example deals with weighted automata. We derive an algorithm due to Stefan Kiefer [12] which he discovered after listening to a presentation of the third author. The third example is that of probabilistic transition systems. We use Gelfand duality to show how to minimize what are called belief automata in the AI literature [11].

2 The Abstract Setting

In this section we describe a categorical setting in which our minimization results can be placed. This framework is by no means exhaustive, we hope it will lend itself to further generalizations and clarifications. In this paper, it merely has the role to present our minimization procedures under a common umbrella. Readers who are interested in concrete examples, rather than in the unifying categorical viewpoint, can skip this section.

The starting point for our presentation of automata minimization is a basic duality: Let \mathcal{C} and \mathcal{D} be dually equivalent categories and let $F : \mathcal{C}^{\text{op}} \rightarrow \mathcal{D}$ and $G : \mathcal{D} \rightarrow \mathcal{C}^{\text{op}}$ be the contravariant functors establishing this duality. The following three examples will be discussed in our paper.

- Example 1.*
1. Let $\mathcal{C} = \mathbf{FinSet}$ be the category of finite sets and $\mathcal{D} = \mathbf{FinBA}$ the category of finite Boolean algebras. The epimorphisms in \mathcal{C} are the surjective maps.
 2. Let $\mathcal{C} = \mathcal{D} = \mathbf{FDVec}$ the category of finite-dimensional real vector spaces and linear maps. The epimorphisms in \mathcal{C} with domain $C \in \mathcal{C}$ correspond to congruence relations on C .
 3. Let $\mathcal{C} = \mathbf{KHaus}$ the category of compact Hausdorff spaces and $\mathcal{D} = \mathbf{C^*Alg}$ be the category of real commutative C^* -algebras. The epimorphisms in \mathcal{C} with domain $C \in \mathcal{C}$ correspond to equivalence relations on C that are closed in the topology.

In our examples we will be dealing with categories \mathcal{C} that have an (Epi,Mono)-factorization system (cf. [2]), i.e., we can factor any morphism $f \in \mathcal{C}$ into $f = m \circ e$ where m is a mono, e is an epi and this factorization is unique up to isomorphism. The automata we are studying in this paper are coalgebras for functors $T : \mathcal{C} \rightarrow \mathcal{C}$ that preserve monos. It is not difficult to see that the factorization structure of \mathcal{C} can be lifted to $\mathbf{Coalg}(T)$ if $T : \mathcal{C} \rightarrow \mathcal{C}$ preserves monos and that the factorization of a morphism in $\mathbf{Coalg}(T)$ can be computed in the base category \mathcal{C} (cf. e.g. [1]). This factorization structure can be used in order to define the minimization of a coalgebra.

Definition 1. Let $T : \mathcal{C} \rightarrow \mathcal{C}$ be a functor and let $S = (S, \gamma)$ be a T -coalgebra. An epimorphism $e : S \rightarrow S'$ of S is called minimization of S iff for all other quotients $e' : S \rightarrow S''$ there exists a unique map $g : S'' \rightarrow S'$ such that $g \circ e' = e$.

In order to ensure that the minimization of a T -coalgebra exists we further assume that the base category is co-wellpowered, i.e., that for every object C in \mathcal{C} the collection of epimorphisms $e : C \rightarrow C'$ forms a set. In this case, given our assumptions on the category \mathcal{C} and the functor $T : \mathcal{C} \rightarrow \mathcal{C}$, the minimization of a coalgebra $(X, \gamma) \in \mathbf{Coalg}(T)$ exists (cf. [1, Thm. 3.8]).

In order to compute the minimization of a T -coalgebra we proceed as follows: We first lift the basic duality $\mathcal{C} \cong \mathcal{D}^{\text{op}}$ to a duality between $\mathbf{Coalg}(T)$ and some category $\mathbf{Alg}(L)$ for some functor $L : \mathcal{D} \rightarrow \mathcal{D}$ and we denote by $\hat{F} : \mathbf{Coalg}(T) \rightarrow \mathbf{Alg}(L)^{\text{op}}$ and $\hat{G} : \mathbf{Alg}(L) \rightarrow \mathbf{Coalg}(T)^{\text{op}}$ the functors that witness this dual equivalence.

Remark 1. Note that a functor $L : \mathcal{D} \rightarrow \mathcal{D}$ such that $\mathbf{Coalg}(T) \cong \mathbf{Alg}(L)^{\text{op}}$ can always be defined as follows: Let $G : \mathcal{D} \rightarrow \mathcal{C}^{\text{op}}$ and $F : \mathcal{C}^{\text{op}} \rightarrow \mathcal{D}$ be the functors that constitute the dual equivalence between \mathcal{C} and \mathcal{D} . If we put $L = F \circ T^{\text{op}} \circ G$, it is not difficult to see that

$\mathbf{Coalg}(\mathbf{T}) \cong \mathbf{Alg}(\mathbf{L})^{\text{op}}$. In our examples we will use, however, more concrete representations of \mathbf{L} and of the algebras in $\mathbf{Alg}(\mathbf{L})$.

By the duality $\mathbf{Coalg}(\mathbf{T}) \cong \mathbf{Alg}(\mathbf{L})^{\text{op}}$ and the fact that minimizations in $\mathbf{Coalg}(\mathbf{T})$ exist it is clear that each object $A \in \mathbf{Alg}(\mathbf{L})$ has a minimal subobject $i : A' \rightarrow A$ such that for all other subobjects $j : A'' \rightarrow A$ there exists a map $k : A' \rightarrow A''$ such that $j \circ k = i$. For any \mathbf{T} -coalgebra (ie, automaton) $\mathcal{S} = (S, \gamma : S \rightarrow \mathbf{T}S)$ the minimization can therefore be obtained by taking the dual of the minimal subobject of the dual algebra $\hat{F}\mathcal{S}$ of \mathcal{S} . We summarize the observations of this section in the following meta-theorem.

Theorem 1. *Let \mathcal{C} be a co-wellpowered category with $(\text{Epi}, \text{Mono})$ -factorization, \mathcal{D} be a category dually equivalent to \mathcal{C} , and $\mathbf{T} : \mathcal{C} \rightarrow \mathcal{C}$ be a functor that preserves monos. Assume also that $\mathbf{L} : \mathcal{D} \rightarrow \mathcal{D}$ is a functor such that $\mathbf{Coalg}(\mathbf{T})$ and $\mathbf{Alg}(\mathbf{L})$ are dually equivalent. Let*

$$\hat{F} : \mathbf{Coalg}(\mathbf{T}) \rightarrow \mathbf{Alg}(\mathbf{L})^{\text{op}} \text{ and } \hat{G} : \mathbf{Alg}(\mathbf{L}) \rightarrow \mathbf{Coalg}(\mathbf{T})^{\text{op}}$$

be the functors establishing this duality. Let \mathcal{S} be any \mathbf{T} -coalgebra with C_m being the minimal subobject of $\hat{F}(\mathcal{S})$. Then $\hat{G}(C_m)$ is the minimization of \mathcal{S} .

3 Partially Observable Deterministic Finite Automata

In this section we are working in the basic setting of Example 1.1.

Definition 2. *We define partially observable deterministic finite automata (PODFA) to be quintuples $\mathcal{S} = (S, \mathcal{A}, O, \delta : S \rightarrow S^{\mathcal{A}}, \gamma : S \rightarrow 2^O)$ where S is a finite set of states, \mathcal{A} is a finite set of actions, O is a finite set of observations, δ is a transition function and γ is an observation function.*

The only difference from the usual automata is the presence of observations. We do not see what state the automaton is in currently, instead we see some observations that partly reveal the state, we may think of γ as a relation between states and observations.

We fix the set of actions and observations henceforth; thus automata are just triples (S, δ, γ) . These automata are coalgebras for the functor $\mathbf{T} : \mathbf{FinSet} \rightarrow \mathbf{FinSet}$ given by

$$\mathbf{T}(S) = S^{\mathcal{A}} \times 2^O, \quad \mathbf{T}(f : S \rightarrow S') = \lambda(\alpha : \mathcal{A} \rightarrow S, O \subseteq O). \langle f \circ \alpha, O \rangle.$$

A homomorphism for these coalgebras is a function $f : S \rightarrow S'$ such that the following diagram commutes:

$$\begin{array}{ccc} S & \xrightarrow{f} & S' \\ \langle \delta, \gamma \rangle \downarrow & & \downarrow \langle \delta', \gamma' \rangle \\ S^{\mathcal{A}} \times 2^O & \xrightarrow{f^{\mathcal{A}} \times \text{id}} & S'^{\mathcal{A}} \times 2^O \end{array}$$

where $f^{\mathcal{A}}(\alpha) = f \circ \alpha$. This translates to the following conditions:

$$\forall s \in S, \omega \in O, \omega \in \gamma(s) \iff \omega \in \gamma'(f(s)), \text{ and } \forall s \in S, a \in \mathcal{A}, f(\delta(s, a)) = \delta'(f(s), a). \quad (1)$$

Definition 3. *The category of coalgebras for the functor \mathbf{T} is called **PODFA**, the category of partially observable deterministic finite automata.*

In order to define a category dual to **PODFA** we use a well-known finite variant of basic Stone duality; see, for example [8].

Fact 2 *The categories **FinSet** and **FinBA** are dually equivalent. This dual equivalence is established via the contravariant functors $2 : \mathbf{FinSet} \rightarrow \mathbf{FinBA}$ and $\text{At} : \mathbf{FinBA} \rightarrow \mathbf{FinSet}$ where 2 denotes the contravariant power set functor and At the functor that maps a given finite Boolean algebra to its set of atoms and a homomorphism $h : B_1 \rightarrow B_2$ to the function $\text{At}(h) : \text{At}(B_2) \rightarrow \text{At}(B_1)$ with $\text{At}(h)(b') := \bigwedge \{b \in B_1 \mid b' \leq h(b)\}$ for all $b' \in \text{At}(B_2)$.*

Based on this duality, we establish a duality between the category **PODFA** and the category of finite Boolean algebras with operators.

Definition 4. *The category **FBAO** of finite Boolean algebras with operators (FBAOs) has as objects finite Boolean algebras B with the usual operators \wedge and \neg with a greatest element \top and least element \perp together with unary operators $(a) : B \rightarrow B$, for each action a , such that (a) is a Boolean homomorphism. For each observation $\omega \in O$, we also have constants $\underline{\omega}$. We denote an object of **FBAO** by*

$$\mathcal{B} = (B, \{(a) \mid a \in \mathcal{A}\}, \{\underline{\omega} \mid \omega \in O\}, \top, \wedge, \neg).$$

*The morphisms of **FBAO** are the usual Boolean homomorphisms preserving, in addition, the constants and the unary operators.*

Remark 2. FBAOs fit easily into the general framework that we outlined in Section 2: We define a functor $L : \mathbf{FinBA} \rightarrow \mathbf{FinBA}$ by putting $LB = \coprod_{a \in \mathcal{A}} B + F_{\mathbf{FinBA}}(O)$ where $F_{\mathbf{FinBA}}(O)$ denotes the free Boolean algebra generated by the finite set of observations O and \coprod and $+$ denote coproducts in **FinBA**. As the category of Boolean algebras is locally finite, finitely generated Boolean algebras are finite and a coproduct of two finite Boolean algebras is also finite. It is easy to see that the categories **Alg**(L) and **FBAO** are isomorphic.

Finite Boolean algebra with operators provide the categorical dual of finite automata.

Proposition 1. $\mathbf{PODFA} \cong \mathbf{FBAO}^{\text{op}}$ via two contravariant functors $\hat{F} : \mathbf{PODFA}^{\text{op}} \rightarrow \mathbf{FBAO}$ and $\hat{G} : \mathbf{FBAO} \rightarrow \mathbf{PODFA}^{\text{op}}$.

Proof. We define the two functors explicitly and check the requisite conditions. First we define a functor $\hat{F} : \mathbf{PODFA}^{\text{op}} \rightarrow \mathbf{FBAO}$ as follows.

$$(S, \delta, \gamma) \mapsto (2^S, \mathbb{A}, \underline{\omega})$$

where $\underline{\omega} = \{s \mid \omega \in \gamma(s)\}$ and $(a)b = \{s \mid \delta(s, a) \in b\}$, b an element of the Boolean algebra 2^S is just a subset of the states of \mathcal{S} . The arrow part of the functor is just inverse image which clearly preserves the usual Boolean operators. The fact that it preserves the new operators and the constants is immediate from equations (1).

Now we define a functor $\hat{G} : \mathbf{FBAO} \rightarrow \mathbf{PODFA}^{\text{op}}$:

$$(B, \mathbb{A}, \underline{\omega}) \mapsto (\text{At}(B), \delta, \gamma).$$

Here $\text{At}(B)$ means the set of atoms of B , since B is a finite Boolean algebra it is certainly atomic. We define

$$\delta(b, a) = \bigwedge \{b' \in B \mid b \leq (a)b'\} \quad (2)$$

and

$$\gamma(b) = \{\omega \in \mathcal{O} \mid b \leq \underline{\omega} \in \mathcal{O}\} \quad (3)$$

where b is an atom of B . The arrow part of the functor is defined as follows:

$$f : B_1 \rightarrow B_2, \quad \hat{G}(f)(b \in \text{At}(B_2)) = \bigwedge \{c \in B_1 \mid b \leq f(c)\}. \quad (4)$$

In order to ensure that \hat{G} is well defined we need to verify that $\delta(b, a)$ is an atom and also that $\hat{G}(f)(b)$ is an atom in the above definitions when b is an atom.

Assume that b is an atom. We claim that $\beta \stackrel{\text{def}}{=}} \bigwedge \{b' \mid b \leq (a)b\} \neq \perp$. Suppose that it were equal to \perp . The set $\{b' \mid b \leq (a)b\}$ consists of finitely many elements $\{b_1, \dots, b_n\}$, so we have $b_1 \wedge b_2 \wedge \dots \wedge b_n = \perp$. Now for each b_i we have $b \leq (a)b_i$ by the definition of the set, so $b \leq (a)b_1 \wedge \dots \wedge (a)b_n$. Using the fact that the operators (a) are Boolean homomorphisms it is easy to see that $b \leq (a)b_1 \wedge \dots \wedge (a)b_n = (a)(b_1 \wedge \dots \wedge b_n) = (a)\perp = \perp$. Thus if $b \neq \perp$ then $\beta \neq \perp$.

In fact there is a unique atom c in the set $\{b' \mid b \leq (a)b\}$. To see that there is an atom in this set we calculate as follows

$$b \leq \top \equiv (a)\top = (a)\left(\bigvee_{x \in \text{At}(B)} x\right) = \bigvee_{x \in \text{At}(B)} (a)x$$

and, since, b is an atom, for some $c \in \text{At}(B)$ we have $b \leq (a)c$. Suppose that there were two such atoms c_1, c_2 . Then $b \leq (a)c_1, b \leq (a)c_2$, hence $b \leq (a)(c_1 \wedge c_2) = (a)\perp = \perp$, which contradicts the assumption that b is an atom. Since $\{b' \mid b \leq (a)b\}$ contains an atom and its meet is not \perp the meet must be that atom. Thus, we could have defined $\delta(b, a)$ to be the unique atom c such that $b \leq (a)c$. The proof that $\hat{G}(f)(b)$ is an atom is similar.

Now suppose that we have a PODFA $\mathcal{S} = (S, \delta, \gamma)$, we construct $\hat{G}(\hat{F}(\mathcal{S}))$ to obtain another PODFA $\mathcal{S}' = (S', \delta', \gamma')$. Clearly the atoms of 2^S just gives back S ; in other words $\{\cdot\} : S \rightarrow \text{At}(2^S)$ is a (natural) isomorphism. Then

$$\gamma'(\{s\}) = \{\omega \in \mathcal{O} \mid \{s\} \subseteq \underline{\omega}\} = \{\omega \mid \omega \in \gamma(s)\} = \gamma(s).$$

Now consider any state s and action a

$$\begin{aligned} \delta'(\{s\}, a) &= \{s'\} && \text{with } \{s\} \subseteq (a)\{s'\} \\ &= \{s'\} && \text{with } \{s\} \subseteq \{s'' \mid \delta(s'', a) = s'\} \\ &= \{\delta(s, a)\}. \end{aligned}$$

Thus the iso in the category of sets is in fact an iso in the category **PODFA**. To show that we get an isomorphism going the other way is straightforward as well. We embed B into $2^{\text{At}(B)}$ by mapping $b \in B$ to $\{u \in \text{At}(B) \mid u \leq b\}$. ■

This shows that we get a precise duality between automata with observations and FBAOs. The interesting point, however, is that one can get a minimal realization via this theory.

Definition 5. Consider the language \mathcal{L} :

$$t ::= \top \mid \hat{\omega} \mid \langle a \rangle t \mid t_1 \wedge t_2 \mid \neg t.$$

Now, for a given automaton $\mathcal{S} = (S, \delta, \gamma)$ we can define a satisfaction relation for the formulas as follows:

$$\begin{aligned} s \models \top & \quad \text{always} & s \models \hat{\omega} & \text{iff } \omega \in \gamma(s) \\ s \models t_1 \wedge t_2 & \text{iff } s \models t_1 \text{ and } s \models t_2 & s \models \neg t & \text{iff } s \not\models t \\ s \models \langle a \rangle t & \text{iff } \delta(s, a) \models t. \end{aligned}$$

We say that a subset U of S is definable by \mathcal{L} if $U = \llbracket t \rrbracket := \{s \in S \mid s \models t\}$ for some $t \in \mathcal{L}$.

From the subsets definable by \mathcal{L} , call this D , we can obtain another Boolean algebra with operators.

Proposition 2. *The set of subsets definable by \mathcal{L} of a PODFA \mathcal{S} , with the evident restrictions of the operators, gives a minimal subalgebra C_m of $\hat{F}(\mathcal{S})$, i.e., a subalgebra C_m that is contained in any other subalgebra of $\hat{F}(\mathcal{S})$.*

The following theorem is an immediate consequence of Theorem 1.

Theorem 3. *The automaton $\hat{G}(C_m)$ is the minimal realization of \mathcal{S} .*

In the previous construction we used a logic that had the Boolean connectives as well as the modal operators and the observations as primitive propositions. In fact, one just needs the last two in the logic provided one is working with deterministic systems. As we will show below in such cases the Boolean connectives are superfluous.

Given the set of actions and observations we define a simple sublanguage \mathcal{L}_0 of \mathcal{L} as follows:

$$\mathcal{L}_0 \ni t ::= \hat{\omega} \mid \langle a \rangle t.$$

The following lemma is easy to prove, if we are working with deterministic automata.

Lemma 1. *If a formula of \mathcal{L} distinguishes two states then so does a formula of \mathcal{L}_0 .*

We say that a set of states is *definable* by \mathcal{L}_0 if it is of the form $\llbracket t \rrbracket$ for some t in \mathcal{L}_0 . The subsets definable by \mathcal{L}_0 do *not* form a Boolean algebra. Let us call the collection of these subsets D' . Then the Boolean algebra generated by D' is exactly the same as the Boolean algebra of the subsets definable by \mathcal{L} . However, this is only true because the operators are required to be boolean algebra homomorphisms. The reason we require that the operators be boolean algebra homomorphisms is because we are dealing with *deterministic* automata. In general, one could have sets defined by formulas like $(a)(p \wedge (b)q)$ in D which are not in the Boolean algebra generated by D' . Thus, we can work with D' and use the Boolean algebra generated by it to get to D and then proceed as we did in the previous subsection.

Proposition 3. *Given a PODFA the boolean algebra with operators generated by the subsets definable by \mathcal{L}_0 gives the zero-generated subobject of the dual FBAO object. Thus one can construct the minimal PODFA by using formulas of \mathcal{L}_0 .*

Our algorithm resembles aspects of Brzozowski's algorithm in the following sense. We construct a dual object which resembles Brzozowski's reversed machine in the sense that an FBAO object is essentially running the original machine backward. However, there are many differences. First of all we are not dealing with reachability aspects and secondly we have multiple observations; essentially we have Moore machines rather than acceptors. The precise categorical version of Brzozowski's algorithm is given by Bonchi et al. [6] who exploit the duality of observability and reachability as originally discussed by Arbib and Manes [3].

4 Linear weighted automata

In this section we show that the minimization of weighted automata fit into our framework. Weighted automata are also called multiplicity automata and are important in learning theory [4]. Our minimization construction is inspired by a construction by Stephan Kiefer [12] who came up with a minimization algorithm after hearing a talk on duality for ordinary automata given by the third author of the present paper. As in [5], we model linear weighted automata as coalgebras for a functor on the category \mathbf{FVect} of finite dimensional real vector spaces. In order to prepare for the categorical picture we recall the well-known self-duality property of the category of finite-dimensional vector spaces.

For a vector space V we denote by $\text{End}(V)$ the set of linear maps $T : V \rightarrow V$ and by V^* the dual space of V consisting of linear mappings $\phi : V \rightarrow \mathbb{R}$. We can easily extend $(\cdot)^*$ to a contravariant functor $(\cdot)^* : \mathbf{FVect} \rightarrow \mathbf{FVect}^{\text{op}}$ that maps a morphism $T : V \rightarrow W$ to the morphism $T^* : W^* \rightarrow V^*$ given by $T^*(\sigma)(v) := \sigma(T(v))$ for $\sigma \in W^*, v \in V$.

Fact 4 *There is a natural isomorphism $\tau : \text{Id}_{\mathbf{FVect}} \rightarrow (\cdot)^{**}$, given by $\tau_V(v)(\phi) = \phi(v)$ for all $v \in V, \phi \in V^*$ and $V \in \mathbf{FVect}$. Therefore the pair $((\cdot)^*, ((\cdot)^*)^{\text{op}})$ is a dual equivalence between \mathbf{FVect} and itself.*

Definition 6. *The category \mathbf{WAut}_o of output linear weighted automata over alphabet \mathcal{A} can be defined as the category of coalgebras for the functor $T = (\cdot)^{\mathcal{A}} \times \mathbb{R}$. Equivalently, an object in \mathbf{WAut}_o can be represented as a triple*

$$S = (V, T : \mathcal{A} \rightarrow \text{End}(V), \eta \in V^*)$$

where $V \in \mathbf{FVect}$, \mathcal{A} is a finite set (input alphabet), and T maps input letters $a \in \mathcal{A}$ to linear transformations $T(a) : V \rightarrow V$. The vector $\eta \in V^*$ is the final vector. The dimension of V is called the dimension of the automaton and is denoted $\dim(S)$. Given a weighted automata $S \in \mathbf{WAut}_o$ we extend the map T to a map $T : \mathcal{A}^* \rightarrow \text{End}(V)$ inductively by putting $T(\epsilon) = \text{id}_V$ and $T(a \cdot w) = T(a) \cdot T(w)$.

Given two linear weighted automata $S_1 = (V_1, T_1 : \mathcal{A} \rightarrow \text{End}(V_1), \eta_1 \in V_1^*)$ and $S_2 = (V_2, T_2 : \mathcal{A} \rightarrow \text{End}(V_2), \eta_2 \in V_2^*)$, a \mathbf{WAut}_o -morphism from S_1 to S_2 is a linear function $M : V_1 \rightarrow V_2$ such that $M^*(\eta_2) = \eta_1$ and for all $a \in \mathcal{A}$ we have $M \circ T_1(a) = T_2(a) \circ M$.

Based on the self-duality of \mathbf{FVect} we can define a duality of output linear weighted automata with linear weighted automata with initial states.

Definition 7. *The category \mathbf{WAut}_i of linear weighted automata with initial state over alphabet \mathcal{A} has as objects triples $(V, T : \mathcal{A} \rightarrow \text{End}(V), \alpha)$ where $V \in \mathbf{FVect}$ is a finite-dimensional real vector space, $T(a) : V \rightarrow V$ is a linear function for all $a \in \mathcal{A}$ and $\alpha \in V$ is an initial state vector. A \mathbf{WAut}_i -morphism from (V_1, T_1, α_1) to (V_2, T_2, α_2) is a linear function $M : V_1 \rightarrow V_2$ such that $M(\alpha_1) = \alpha_2$ and such that for all $a \in \mathcal{A}$ we have $M \circ T_1(a) = T_2(a) \circ M$.*

Remark 3. Again the dual category \mathbf{WAut}_i can be represented as a category of algebras for a functor. We define a functor $L : \mathbf{FVect} \rightarrow \mathbf{FVect}$ by putting $LV = \coprod_{a \in \mathcal{A}} V + F_{\mathbf{FVect}}(1)$ where $F_{\mathbf{FVect}}(1) = \mathbb{R}$ denotes the free real vector space generated by the one-element set. It is now not difficult to see that $\mathbf{Alg}(L)$ is isomorphic to \mathbf{WAut}_i .

We now use the duality from Fact 4 to define the dual of a linear weighted automaton in order to establish a duality $\mathbf{WAut}_o \cong \mathbf{WAut}_i^{\text{op}}$.

Definition 8. Let $\mathcal{S} = (V, T, \eta)$ be an output linear weighted automaton. The dual of \mathcal{S} is defined as $\hat{F}(\mathcal{S}) = (V^*, T^*, \eta)$ where $T^*(w) := (T(w))^*$ for $w \in \mathcal{A}^*$. We extend \hat{F} to a contravariant functor from \mathbf{WAut}_o to \mathbf{WAut}_i by putting $\hat{F}(M) := M^*$ for a given morphism $M : \mathcal{S}_1 \rightarrow \mathcal{S}_2 \in \mathbf{WAut}_o$. Likewise, the dual of a linear weighted automaton with initial state $\mathcal{S} = (V, T, \alpha)$ is defined as $\hat{G}(\mathcal{S}) = (V^*, T^*, \tau_V(\alpha))$ where $\tau_V : V \rightarrow V^{**}$ is the V -component of the natural isomorphism from Fact 4 and we extend \hat{G} to a contravariant functor from \mathbf{WAut}_i to \mathbf{WAut}_o by putting $\hat{G}(M) = M^*$.

Proposition 4. The natural isomorphism $\tau : \text{Id}_{\mathbf{FVect}} \rightarrow (\cdot)^{**}$ extends to natural isomorphisms $\tau : \text{Id}_{\mathbf{WAut}_i} \rightarrow \hat{F} \circ \hat{G}$ and $\tau : \text{Id}_{\mathbf{WAut}_o} \rightarrow \hat{G} \circ \hat{F}$. Consequently, the contravariant functors $\hat{F} : \mathbf{WAut}_o \rightarrow \mathbf{WAut}_i$ and $\hat{G} : \mathbf{WAut}_i \rightarrow \mathbf{WAut}_o$ form a dual equivalence of \mathbf{WAut}_o and \mathbf{WAut}_i .

Proof. We show that for any linear weighted automaton $\mathcal{S} = (V, T : \mathcal{A}^* \rightarrow \text{End}(V), \alpha \in V, \eta \in V^*)$ the map $\tau_V : V \rightarrow V^{**}$ is an isomorphism between \mathcal{S} and $\hat{G}\hat{F}(\mathcal{S})$. Clearly it is an isomorphism of the underlying vector spaces.

We need to check that $\tau_V : \mathcal{S} \rightarrow \hat{G}\hat{F}(\mathcal{S}) \in \mathbf{WAut}_o$, where

$$\mathcal{S}^{dd} = (V^{**}, T^{**}, \tau_{V^*}(\eta))$$

by definition.

Obviously τ_V satisfies the morphism condition for the initial vector. For the condition regarding the final vector, one can easily see that

$$\begin{aligned} \tau_V^*(\tau_{V^*}(\eta))(v) &\stackrel{(\text{Def. of } (\cdot)^*)}{=} (\tau_{V^*}(\eta))(\tau_V(v)) \\ &\stackrel{(\text{Def. of } \tau_{V^*})}{=} \tau_V(v)(\eta) \\ &\stackrel{(\text{Def. of } \tau_V)}{=} \eta(v) \end{aligned}$$

for any $v \in V$. Therefore $\tau_V^*(\tau_{V^*}(\eta)) = \eta$ as required. Furthermore, for $w \in \mathcal{A}^*$, $v \in V$ and $\phi \in V^*$, we calculate:

$$\begin{aligned} (T^{**}(w)(\tau_V(v)))(\phi) &= ((T^*(w))^*(\tau_V(v)))(\phi) \\ &= \tau_V(T(w)^*(\phi)) = (T(w)^*(\phi))(v) \\ &= \phi(T(w)(v)) = (\tau_V(T(w)(v)))(\phi) \end{aligned}$$

which implies that $T^{**}(w)(\tau_V(v)) = \tau_V(T(w)(v))$ as required by the diagram in the definition of a \mathbf{WAut}_o -morphism. Similarly one can check that τ gives rise to a natural isomorphism from $\text{Id}_{\mathbf{WAut}_i}$ to $\hat{F} \circ \hat{G}$. \blacksquare

In order to apply our general minimization theorem we have to describe the minimal sub-objects of an object $\mathcal{S} \in \mathbf{WAut}_i$. As in the previous case this can be done using formulas. One can think of these formulas as *tests* which can be applied to a given weighted automaton.

Definition 9. Let \mathcal{A} be a finite alphabet. The set of tests \mathcal{T} over \mathcal{A} is inductively defined by

$$\mathcal{T} \ni t ::= \epsilon \mid a \cdot t, a \in \mathcal{A}.$$

Given some $\mathcal{S} = (V, T, \eta \in V^*) \in \mathbf{WAut}_o$ the semantics of a test $t \in \mathcal{T}$ is a linear function $\llbracket t \rrbracket : V \rightarrow \mathbb{R}$ that is defined by putting $\llbracket \epsilon \rrbracket(v) = \eta(v)$ and $\llbracket a \cdot t \rrbracket(v) = \llbracket t \rrbracket(T(a)(v))$.

We can use these test functions in order to define elements of $\hat{F}(S)$ and in order to obtain a way to compute minimal subobjects in the category \mathbf{WAut}_i :

Lemma 2. *Let $S = (V, T, \eta) \in \mathbf{WAut}_o$ and let $\hat{F}(S) = (V^*, T^*, \eta)$. The smallest subobject of $\hat{F}(S)$ is equal to $(U, T^*_{\upharpoonright U}, \eta)$ where $U = \text{span}(\{\llbracket t \rrbracket \mid t \in \mathcal{T}\})$ denotes the subspace of V generated from test functions and $T^*_{\upharpoonright U}$ denotes the restriction of T^* to U , ie., $T^*_{\upharpoonright U}(a) = (T(a)^*)_{\upharpoonright U}$.*

Proof. In order to see that $T^*_{\upharpoonright U}$ is well-defined we have to check that for any test $t \in \mathcal{T}$ and any $a \in \mathcal{A}$ we have $T(a)^*(\llbracket t \rrbracket) \in U$. For $v \in V$ we have $T(a)^*(\llbracket t \rrbracket)(v) = \llbracket t \rrbracket(T(a)(v)) = \llbracket a \cdot t \rrbracket(v)$ and thus $T(a)^*(\llbracket t \rrbracket) = \llbracket a \cdot t \rrbracket \in U$ as required. It is not difficult to see that any subobject of $\hat{F}(S)$ has to contain $\{\llbracket t \rrbracket \mid t \in \mathcal{T}\}$ and therefore also U , which shows that $(U, T^*_{\upharpoonright U}, \eta)$ is indeed the smallest subobject of $\hat{F}(S)$. ■

By Theorem 1 the following is immediate:

Theorem 5. *Let $S = (V, T, \eta) \in \mathbf{WAut}_o$ be an output linear weighted automaton. The minimization of S can be computed as $\hat{G}(S')$ where S' is the minimal subobject of $\hat{F}(S) \in \mathbf{WAut}_i$ as described in Lemma 2.*

Spelling out the description of minimal subobjects provided by Lemma 2 one can now obtain a minimization procedure for linear weighted automata. Owing to space limitations, however, we cannot provide the details of this construction here. Finally note that in [5] it is proven that behavioural equivalence in the category that we call \mathbf{WAut}_o is precisely language equivalence. This shows that minimization in \mathbf{WAut}_o coincides with the standard notion of minimization for linear weighted automata.

5 Belief automata

A fundamental example of our approach is probabilistic transition systems. These are not a special case of weighted automata for the simple reason that the set of probability distributions on a set does not form a vector space. One needs a different theory. There are two possible approaches: one is to use Stone's version of Gelfand duality [18, 19] and the other is to exploit convexity ideas, which we will not do here but will follow up in future work. For textbook presentations of Stone's version of Gelfand duality we recommend Johnstone [10]. Some key categorical constructions are due to Joan Wick-Pelletier [14].

There are several versions of probabilistic transition systems that one could use. A basic model is *partially observable Markov decision processes* (POMDPs) [17, 16]. This is a model invented in the operations research community but now of major importance in the artificial intelligence community, especially in machine learning; see, for example, the paper by Kaelbling et al. [11]. Usually POMDPs have a notion of *reward* associated with transitions and the main interest is in optimizing the reward by finding a suitable policy. However, we will ignore the reward for this paper. POMDPs without rewards are rather like Labelled Markov Processes [15] except that the state is not observable. A very interesting duality theory for LMPs also based on Gelfand duality has been developed by Mislove et al. [13]. Our constructions are somewhat different because we deal with partial observability. We use the left adjoint of a particular forgetful functor to construct C^* -algebras [14], they construct C^* -algebras directly

from tests. They use their theory to discuss composition of LMPs while we are interested in minimization.

In the AI literature one does not deal with POMDPs directly, rather one deals with what are called belief automata. This is an automaton where the states are probability distributions over the states of the original POMDP and the transitions are interpreted as changes in the distribution. One works directly with this object for analysis, learning prediction etc. and the original POMDP is not used. The belief automaton becomes a set of probability distributions over a finite set which is just a simplex in a finite dimensional real vector space and hence a compact Hausdorff space.

Definition 10. We define $\mathbf{CHA}(\mathcal{A}, O)$ as the category of compact Hausdorff automata over \mathcal{A} and O . The objects are triples $\mathcal{K} = (K, \Delta : K \rightarrow K^{\mathcal{A}}, \Gamma : K \rightarrow \text{Sub}(O))$, where K is a compact Hausdorff space and Δ is a transition function over the set of actions \mathcal{A} while Γ maps a state to a subdistribution over the set of observations O , ie. to a function $f : O \rightarrow [0, 1]$ such that $\sum_{\omega \in O} f(\omega) \leq 1$. For fixed $a \in \mathcal{A}$, the function $\Delta(\cdot)(a) : K \rightarrow K$ is continuous and Γ is continuous when $\text{Sub}(O)$, the set of subdistributions over O , is topologized with the topology it inherits as a subset of $[0, 1]^{|O|}$.

The morphisms of $\mathbf{CHA}(\mathcal{A}, O)$, $f : (K, \Delta, \Gamma) \rightarrow (K', \Delta', \Gamma')$ are continuous functions $f : K \rightarrow K'$ such that $\Delta'(f(k))(a) = f(\Delta(k)(a))$ and $\Gamma'(f(k))(\omega) = \Gamma(k)(\omega)$ for $k \in K$, $a \in \mathcal{A}$ and $\omega \in O$. In other words, $\mathbf{CHA}(\mathcal{A}, O)$ is isomorphic to the category of coalgebras for the functor $T = (-)^{\mathcal{A}} \times \text{Sub}(O) : \mathbf{KHaus} \rightarrow \mathbf{KHaus}$ on the category of compact Hausdorff spaces.

We will usually just write \mathbf{CHA} instead of $\mathbf{CHA}(\mathcal{A}, O)$. We employ the well known Gelfand duality between the category of compact Hausdorff spaces and the category $\mathbf{C}^*\mathbf{Alg}$ of real commutative C^* -algebras in order to develop a duality for \mathbf{CHA} (cf. [10, Chapter IV.4]). For the dual of a compact Hausdorff automaton we take commutative C^* -algebras as the underlying structure but we need to equip them with operators as we did with Boolean algebras with operators. We will assume fixed sets \mathcal{A} of actions and O of observations.

Definition 11. An object of the category \mathbf{CAO} is a real commutative C^* -algebra C together with a set of operators $(a) : C \rightarrow C$, for each $a \in A$ and a distinguished set of elements $\underline{\omega}$ of C for each $\omega \in O$. The operators (a) are C^* -algebra morphisms (ie. ring homomorphisms). We require that $0 \leq \underline{\omega} \leq 1$ and $\sum_{\omega \in O} \underline{\omega} \leq 1$. The morphisms are morphisms of C^* -algebras, which also preserve the constants and the additional operators.

Remark 4. Again it is possible to view \mathbf{CAO} as a category L-algebras. There is, however, a slight problem that needs to be solved: the forgetful functor from the category of C^* -algebras does not have a left adjoint, ie., it is not possible to always construct the free C^* -algebra for a given set of generators. Luckily there is a slight variant of the forgetful functor that does have a left adjoint: Consider the functor from $\mathbf{C}^*\mathbf{Alg}$ to \mathbf{Set} that maps a given C^* -algebra C to its unit interval $I = \{c \in C \mid 0 \leq c \leq 1\}$. This functor does have a left-adjoint ([14]) which we denote with $F_{\mathbf{C}^*\mathbf{Alg}}$. We now define a functor $L : \mathbf{C}^*\mathbf{Alg} \rightarrow \mathbf{C}^*\mathbf{Alg}$ by putting $LC = \coprod_{a \in \mathcal{A}} C + F_{\mathbf{C}^*\mathbf{Alg}}(O)/J$ where $F_{\mathbf{C}^*\mathbf{Alg}}(O)/J$ is a quotient of $F_{\mathbf{C}^*\mathbf{Alg}}(O)$ that ensures that the sum of the elements of $F_{\mathbf{C}^*\mathbf{Alg}}(O)/J$ that correspond to the constants ω is smaller or equal to 1. A number of details have to be checked; owing to space limitations we omit them here and will give them in the full version of the paper.

There are contravariant functors between these categories that establish a dual equivalence. We describe these functors before stating the duality theorem. We name the functors $\mathbf{A} : \mathbf{CHA} \rightarrow \mathbf{CAO}^{\text{op}}$ and $\mathbf{H} : \mathbf{CAO}^{\text{op}} \rightarrow \mathbf{CHA}$.

We are given an object $\mathcal{K} = (K, \Delta, \Gamma)$ of \mathbf{CHA} . We define an object $\mathbf{A}(\mathcal{K}) = (C, \{(a)\}, \{\underline{\omega}\})$ of \mathbf{CAO} as follows. We define C to be the C^* -algebra of continuous functions from K to \mathbb{R} with the pointwise order.

The operators and constants are defined by $(a)f := \lambda x : K.f(\Delta(x)(a))$ for $a \in \mathcal{A}$ and $f \in C$ and $\underline{\omega} := \lambda x : K.\Gamma(x)(\omega)$ for $\omega \in \mathcal{O}$. Clearly $(a) : C \rightarrow C$ is a ring homomorphism for any $a \in \mathcal{A}$. Also,

$$\sum_{\omega \in \mathcal{O}} \underline{\omega} = \sum_{\omega \in \mathcal{O}} \lambda x : K.\Gamma(x)(\omega) \leq \lambda x : K.1.$$

Given a morphism $h : \mathcal{K} \rightarrow \mathcal{K}'$, where $\mathcal{K}' = (K', \Delta', \Gamma')$ we define $\mathbf{A}(h) : \mathbf{A}(\mathcal{K}') \rightarrow \mathbf{A}(\mathcal{K})$ by $\mathbf{A}(h)(f) = f \circ h$. It is easy to verify that this is a functor.

For the reverse direction we proceed as follows: Given a \mathbf{CAO} $(C, \{(a)\}, \{\underline{\omega}\})$ we define the dual compact Hausdorff automaton $\mathcal{K} = (K, \Delta, \Gamma)$ by putting $K = \text{Hom}(C, \mathbb{R})$, $\Delta(f)(a) = \lambda x.f((a)x)$ and $\Gamma(f)(\omega) = f(\underline{\omega})$ for $a \in \mathcal{A}$, $\omega \in \mathcal{O}$ and $f \in \text{Hom}(C, \mathbb{R})$. The set K is turned into a compact Hausdorff space by adding the topology that is generated by the sets $\hat{c} = \{f \in \text{Hom}(C, \mathbb{R}) \mid f(c) \neq 0\}$.

Lemma 3. *With these definitions the maps Δ and Γ are continuous.*

Proof. It is not difficult to see that Δ is continuous:

$$\Delta_a^{-1}(\hat{c}) = \{g \mid \Delta_a(g)(c) \neq 0\} = \{g \mid g((a)(c)) \neq 0\} = \widehat{(a)(c)}.$$

The fact that Γ is well-defined follows because any C^* -algebra morphism $f \in \text{Hom}(C, \mathbb{R})$ has the property that $\|f(x)\| \leq \|x\|$ for all $x \in C$ and thus $\|\sum_{\omega \in \mathcal{O}} f(\underline{\omega})\| \leq \sum_{\omega \in \mathcal{O}} \|\underline{\omega}\| \leq 1$. The fact that Γ is continuous can be seen as follows: By Gelfand duality we can assume w.l.o.g. that $C \cong C(X)$ for some compact Hausdorff space X . Also by Gelfand duality we know that the map

$$\begin{aligned} \tau_X : X &\rightarrow (C(X) \rightarrow \mathbb{R}) \\ x &\mapsto \lambda f.f(x) \end{aligned}$$

is a homeomorphism. In particular, for any $g : C(X) \rightarrow \mathbb{R}$ there is $x_g \in X$ such that $\tau_X(x_g) = g$. Our goal is to prove that the map

$$\begin{aligned} \Gamma_\omega : \mathcal{HC}(X) &\rightarrow \mathbb{R} \\ (g : C(X) \rightarrow \mathbb{R}) &\mapsto g(\underline{\omega}) \end{aligned}$$

is continuous. Let $U \subseteq \mathbb{R}$ be an open set. Then

$$\begin{aligned} \Gamma_\omega^{-1}(U) &= \{g : C(X) \rightarrow \mathbb{R} \mid g(\underline{\omega}) \in U\} \\ &= \tau_X(\{x \mid \underline{\omega}(x) \in U\}) = \tau_X(\underline{\omega}^{-1}(U)) \end{aligned}$$

Because $\underline{\omega} \in C(X)$ is continuous we have $\underline{\omega}^{-1}(U)$ is open and by the fact that τ_X is a homeomorphism we have $\Gamma_\omega^{-1}(U)$ is open as well. As $U \subseteq \mathbb{R}$ was an arbitrary open set this shows that Γ_ω is continuous as required. \blacksquare

This operation can be extended to a functor $\mathbf{H} : \mathbf{CAO} \rightarrow \mathbf{CHA}^{\text{op}}$ that maps a function $h : C_1 \rightarrow C_2 \in \mathbf{CAO}$ to the function $\mathbf{H}(h) : \mathbf{H}(C_2) \rightarrow \mathbf{H}(C_1)$ given by $\mathbf{H}(h)(g) := g \circ h$.

Theorem 6. *The functors $\mathbf{H} : \mathbf{CAO} \rightarrow \mathbf{CHA}^{\text{op}}$ and $\mathbf{A} : \mathbf{CHA}^{\text{op}} \rightarrow \mathbf{CAO}$ form an equivalence of categories, i.e. the categories \mathbf{CHA} and \mathbf{CAO} are dually equivalent: $\mathbf{CHA} \simeq \mathbf{CAO}^{\text{op}}$.*

Proof. Let $\mathcal{K} = (K, \Delta, \Gamma)$ be a compact Hausdorff automaton and let $\mathbf{H}\mathcal{K} = (K', \Delta', \Gamma')$. It is an immediate consequence of Gelfand duality that the carrier spaces K and K' are isomorphic via the isomorphism $\tau_{\mathcal{K}} : K \rightarrow K'$ given by $\tau_{\mathcal{K}}(x)(f) := f(x)$ (note that $K' = \text{Hom}(C(K), \mathbb{R})$, i.e. $\tau_{\mathcal{K}}(x)$ has to map continuous real-valued functions on K to a real number). The fact that $\tau_{\mathcal{K}} : \text{Id} \rightarrow \mathbf{H} \circ \mathbf{A}$ is natural is also a consequence of Gelfand duality. The only thing which remains to be checked is that $\tau_{\mathcal{K}}$ is an automaton morphism. In order to see this we calculate for $a \in \mathcal{A}$, $x \in K$ and $g \in C(K)$ that

$$\begin{aligned} (\Delta'(\tau_{\mathcal{K}}(x))(a))(g) &= \tau_{\mathcal{K}}(x)((a)g) = \tau_{\mathcal{K}}(x)(\lambda y. g(\Delta(y)(a))) \\ &= g(\Delta(x)(a)) = \tau_{\mathcal{K}}(\Delta(x)(a))(g) \end{aligned}$$

where all equalities are immediate consequences of the definitions. This shows that $\tau_{\mathcal{K}}$ is an automaton morphism regarding the transition structures Δ and Δ' . Let us now check that $\tau_{\mathcal{K}}$ also preserves all observations $\omega \in \mathcal{O}$:

$$\Gamma'(\tau_{\mathcal{K}}(x))(\omega) = \tau_{\mathcal{K}}(x)(\underline{\omega}) = \underline{\omega}(x) = \Gamma(x)(\omega).$$

For the other direction of the equivalence we have to show that the C^* -algebra isomorphism $\alpha_C : C \rightarrow C(\text{Hom}(C, \mathbb{R}))$ that exists by Gelfand duality for each $C = (C, \{(a)\}, \{\underline{\omega}\}) \in \mathbf{CAO}$ and that is given by $\alpha_C(c)(f) := f(c)$ for $f \in \text{Hom}(C, \mathbb{R})$ is in fact a \mathbf{CAO} -morphism. It then follows that $\alpha : \text{Id} \rightarrow \mathbf{A} \circ \mathbf{H}$ is a natural isomorphism. In order to see that α_C is indeed a \mathbf{CAO} -morphism we calculate

$$\begin{aligned} (\alpha_C((a)c)(f)) &= f((a)c) = (\Delta(f)(a))(c) \\ &= \alpha_C(c)(\Delta(f)(a)) = (a)'(\alpha_C(c)) \\ (\alpha_C(\underline{\omega}))(f) &= \underline{\omega}(f) = \Gamma(f)(\omega) = \underline{\omega}'(f) \end{aligned}$$

where the $(a)'$ and $\underline{\omega}'$ denote the operators and constants of \mathbf{AHC} which is the C^* -algebra with operators obtained by following both functors. \blacksquare

We are now going to use the duality in order to minimize compact Hausdorff automata. Let us first define a set of tests.

Definition 12. *The set of tests for automata in \mathbf{CHA} is defined as follows*

$$\mathcal{T} \ni t ::= \underline{\omega}, \omega \in \mathcal{O} \mid (a)t, a \in \mathcal{A}.$$

A test t gives rise to a function $\llbracket t \rrbracket : K \rightarrow \mathbb{R}$:

$$\begin{aligned} \llbracket \underline{\omega} \rrbracket(k) &:= \Gamma(k)(\omega) && \text{for } \omega \in \mathcal{O} \\ \llbracket (a)t \rrbracket(k) &:= \llbracket t \rrbracket(\Delta(k)(a)) && \text{for } \omega \in \mathcal{O}, w \in \mathcal{A}^*. \end{aligned}$$

We say two states $k, k' \in K$ in an automaton (K, Δ, Γ) are test equivalent (notation: $k \sim k'$) if for all tests $t \in \mathcal{T}$ we have $\llbracket t \rrbracket(k) = \llbracket t \rrbracket(k')$.

The tests can be used in order to obtain a better understanding of the smallest subobject (subalgebra) of a given C^* -algebra with operators $\mathbf{A}(\mathcal{K}) \in \mathbf{CAO}$.

Proposition 5. *Let $\mathcal{K} \in \mathbf{CHA}$ be a compact Hausdorff automaton, let $\mathbf{A}(\mathcal{K})$ be its dual C^* -algebra with operators and let $C = (C, \{(a)\}, \{\omega\})$ be the smallest subobject (subalgebra) of $\mathbf{A}(\mathcal{K})$. Then $C \subseteq \{f \in C(K) \mid \forall k, k'. k \sim k' \text{ implies } f(k) = f(k')\}$, ie. all functions in C agree on the equivalence classes defined by test equivalence on \mathcal{K} .*

Proof.(Sketch) It is clear that $\llbracket \mathcal{T} \rrbracket = \{\llbracket t \rrbracket \mid t \in \mathcal{T}\} \subseteq C$. We can generate a C^* -algebra with operators from $\llbracket \mathcal{T} \rrbracket$ by adding the constant functions and by closing under $+$, \cdot and under limits in the sup norm. All functions generated in that way, and thus all functions in C , will be constant on the \sim -equivalence classes. ■

As in the other cases duality gives us the following theorem.

Theorem 7. *Let $\mathcal{K} \in \mathbf{CHA}$ be a compact Hausdorff automaton. The minimization of \mathcal{K} can be computed as $\mathbf{H}(C)$ where C is the minimal subobject of $\mathbf{A}(\mathcal{K}) \in \mathbf{CAO}$ as described in Proposition 5.*

As a corollary of the previous proposition and theorem we obtain a description of the minimization of a given compact Hausdorff automaton.

Corollary 1. *Given a compact Hausdorff automaton $\mathcal{K} = (K, \Delta, \Gamma)$, let $\mathcal{K}' = (K, \Delta', \Gamma')$ be a compact Hausdorff automaton such that the states of \mathcal{K}' are \sim -equivalence classes of \mathcal{K} and $\Delta'([k])(a) = [\Delta(k)(a)]$, $\Gamma'([k]) = \Gamma(k)$ for $k \in K$ where $[k]$ denotes the \sim -equivalence class of k . Then \mathcal{K}' is the minimization of \mathcal{K} .*

6 Conclusions

In this paper we have developed a theory of minimization via duality and showed how it works in three examples. There are some variations of these cases that we have omitted from this abstract for lack of space. For example, we have worked out a version of the weighted automaton case in which there are both initial and final “states” and in which the minimization is achieved by dualizing and using reachability twice. This version is actually being used by Stefan Kiefer and he has implemented the algorithm.

The idea of this kind of double duality was first worked out, in a completely non-categorical way in 2006 [9] and was used as a way of representing systems where one does not know the state space. Closely connected to what we have been doing is the work in [6]: they give a duality-based explanation of the Brzozowski algorithm for deterministic finite automata, but their duality is very different from our Stone duality and takes reachability into account, something we have not done here.

We note that Mislove et al. [13] have used this duality to study labelled Markov processes but with different motivations; their work, while very interesting, is not about minimization but about how to compose systems. Furthermore, the details of their work are somewhat different from ours.

It is clear that there are a number of cases very close to automata, for example tree automata, nondeterministic automata, alternating automata etc. that fall under the same rubric and could

be developed as straightforward extensions of the present work. We have begun a collaboration with the authors of [6] and others to unify our points of view and collect several more examples.

Acknowledgments

We have benefitted from discussions with Marcello Bonsangue, Marcelo Fiore, Dexter Kozen, Alexander Kurz, Mike Mislove, Robert Myers, Jan Rutten, Alexandra Silva and James Worrell on this and related topics. We are very grateful to Vincenzo Marra who shared with us his expert knowledge on C^* -algebras and to Stefan Kiefer for explaining to us his algorithm for minimizing weighted automata. The third author would like to thank Chris Hundt, Monica Dinculescu, Joelle Pineau and Doina Precup for the collaboration that led to the precursor of this paper.

This research is supported by a grant from the Natural Sciences and Engineering Research Council of Canada to the third author. The second author acknowledges support by the EU FP7 project SEALS and by the EPSRC projects ConDOR, ExODA and LogMap. The work of the first author was partially supported by the EPSRC grants EP/F032102/1 and EP/F031173/1 and by Rustaveli Science Foundation of Georgia grant FR/489/5-105/11.

References

1. Adámek, J., Bonchi, F., Hülsbusch, M., König, B., Milius, S., Silva, A.: A coalgebraic perspective on minimization and determinization. In: Proceedings of the Fifteenth International Conference on Foundations of Software Science and Computation structures (FoSSaCS 2012). Lecture Notes in Computer Science (2012)
2. Adamek, J., Herrlich, H., Strecker, G.E.: Abstract and Concrete Categories: The Joy of Cats. Dover (1990)
3. Arbib, M., Manes, E.: Adjoint machines, state behavior machines and duality. *J. Pure Appl. Algebra* 6, 313–343 (1975)
4. Beimel, A., Bergadano, F., Bshouty, N.H., Kushelevitz, E., Varricchio, S.: Learning functions represented as multiplicity automata. *Journal of the ACM* 47(5), 506–530 (May 2000)
5. Bonchi, F., Bonsangue, M., Boreale, M., Rutten, J., Silva, A.: A coalgebraic perspective on linear weighted automata. *Information and Computation* 211, 77–105 (2012)
6. Bonchi, F., Bonsangue, M., Rutten, J., Silva, A.: Brzozowski’s algorithm (co)algebraically. In: Constable, R., Silva, A. (eds.) *Logics and Program Semantics: Essays Dedicated to Dexter Kozen*. Lecture Notes In Computer Science, vol. 7230, pp. 12–23. Springer-Verlag (2012)
7. Brzozowski, J.A.: Canonical regular expressions and minimal state graphs for definite events. In: Fox, J. (ed.) *Proceedings of the Symposium on Mathematical Theory of Automata*, pp. 529–561. No. 12 in MRI Symposia Series, Polytechnic Press of the Polytechnic Institute of Brooklyn (April 1962), book appeared in 1963.
8. Givant, S., Halmos, P.: *Introduction to Boolean Algebras*. Undergraduate Texts in Mathematics, Springer-Verlag (2009)
9. Hundt, C., Panangaden, P., Pineau, J., Precup, D.: Representing systems with hidden state. In: *The Twenty-First National Conference on Artificial Intelligence (AAAI)* (2006)
10. Johnstone, P.: *Stone Spaces*, Cambridge Studies in Advanced Mathematics, vol. 3. Cambridge University Press (1982)
11. Kaelbling, L.P., Littman, M.L., Cassandra, A.R.: Planning and acting in partially observable stochastic domains. *Artificial Intelligence* 101 (1998)
12. Kiefer, S.: Minimization of weighted automata (2011), unpublished private communication

13. Mislove, M., Ouaknine, J., Pavlovic, D., Worrell, J.: Duality for labelled Markov processes. In: Walukiewicz, I. (ed.) Foundations of Software Science and Computation Structures, FOSSACS. Lecture Notes In Computer Science, vol. 2987, pp. 393–407 (2004)
14. Negrepointis, J.W.: Duality in analysis from the point of view of triples. *Journal of Algebra* 19, 228–253 (1971)
15. Panangaden, P.: *Labelled Markov Processes*. Imperial College Press (2009)
16. Smallwood, R.D., Sondik, E.J.: The optimal control of partially observable Markov decision processes over a finite horizon. *Operations Research* 21(5), 1071–1088 (1973)
17. Sondik, E.J.: The optimal control of partially observable Markov processes. Ph.D. thesis, Stanford University (1971)
18. Stone, M.H.: A general theory of spectra I. *Proc. Nat. Acad. Sci. USA* 26, 280–283 (1940)
19. Stone, M.H.: A general theory of spectra II. *Proc. Nat. Acad. Sci. USA* 27, 83–87 (1941)